МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПЕЧАТИ

имени Ивана Федорова»

Лабораторная работа

«Разработка галереи на языке Perl»

по дисциплине

«Программирование на языке Perl»

Исполнитель:

Попкова Е.С.

Группа: Дцис 5-1

Руководитель:

Ноздрачев С.А.

Дата сдачи:

Подпись руководителя:

Москва 2014

СОДЕРЖАНИЕ

1 Техническое задание. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2 Ход выполнения работы . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

3 Алгоритм работы приложения и блок-схема ИС . . . . . . . . . . . . . . . . 5

4 Скриншоты работы информационной системы . . . . . . . . . . . . . . . . . 7

Приложение А. Исходные коды ИС. . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

1 Техническое задание

На языке Perl с использованием технологии ajax необходимо разработать фотогалерею. Фотографии и их описания будут храниться в базе данных. Пользователю должна быть доступна загрузка, просмотр и удаление фотографий, редактирование описаний.

2 Ход выполнения работы

Для написания приложения использовался язык программирования Perl. Для доступа к базе данных были подключены модули DBI и DBD::mysql. База данных представляет собой одну таблицу photo(id, src, desc), где src-путь до изображения, а desc-его описание. Данная система представлена несколькими .pl файлами: gallery.pl, где происходит вывод основной html-разметки, и в которой подключены основные библиотеки и остальные .pl файлы; updel.pl – код удаления изображений и редактирования описаний к изображениям; up.pl – код загрузки изображений; showGallery.pl – вывод главного изображения и миниатюр; config.pl – подключение к базе данных.

3 Алгоритм работы приложения и блок-схема ИС

1) Подключение необходимых библиотек для работы с БД

2) Если получена POST-переменная с id изображения, то пункт 3

3) Удаляем изображение из БД

4) Если получена POST-переменная с описанием изображения, то сохраняем изменение описания в БД

5) Сохраняем в переменную путь до папки с изображениями

6) Задаем максимальный размер файла

7) Если POST-переменная action равна значению upload, то пункты 8-16

8) Формируем путь до нового изображения вместе с названием файла

9) Определяем максимальный размер файла

10) Если такой файл еще не существует, то пункты 11-15

11) Открываем файл

12) Читаем файл и записываем его

13) Закрываем файл

14) Получаем размер загруженного файла

15) Если размер файл больше заданного максимального, то удаляем файл, показываем ошибку

16) Вставляем информацию о файле в базу данных

17) Выводим галерею и миниатюры



Рисунок 1 – Блок-схема ИС

4 Скриншоты работы информационной системы

На рисунке 2 показан вид галереи.

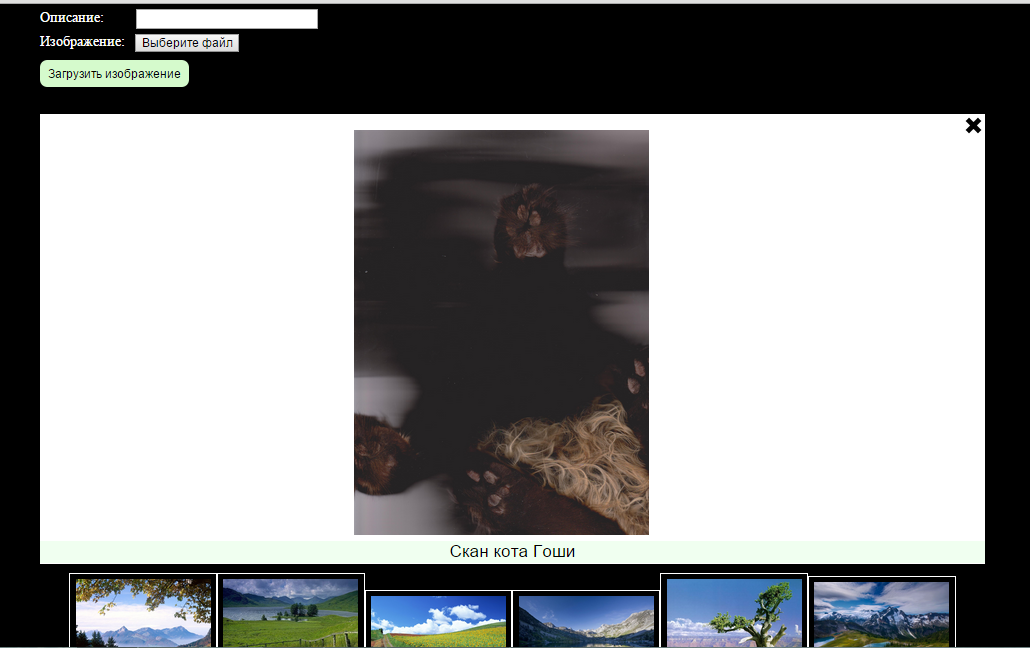


Рисунок 2 – Главная фотография и миниатюры

На рисунке 3 изображен вид галереи при изменении описания изображения или при удалении изображения.

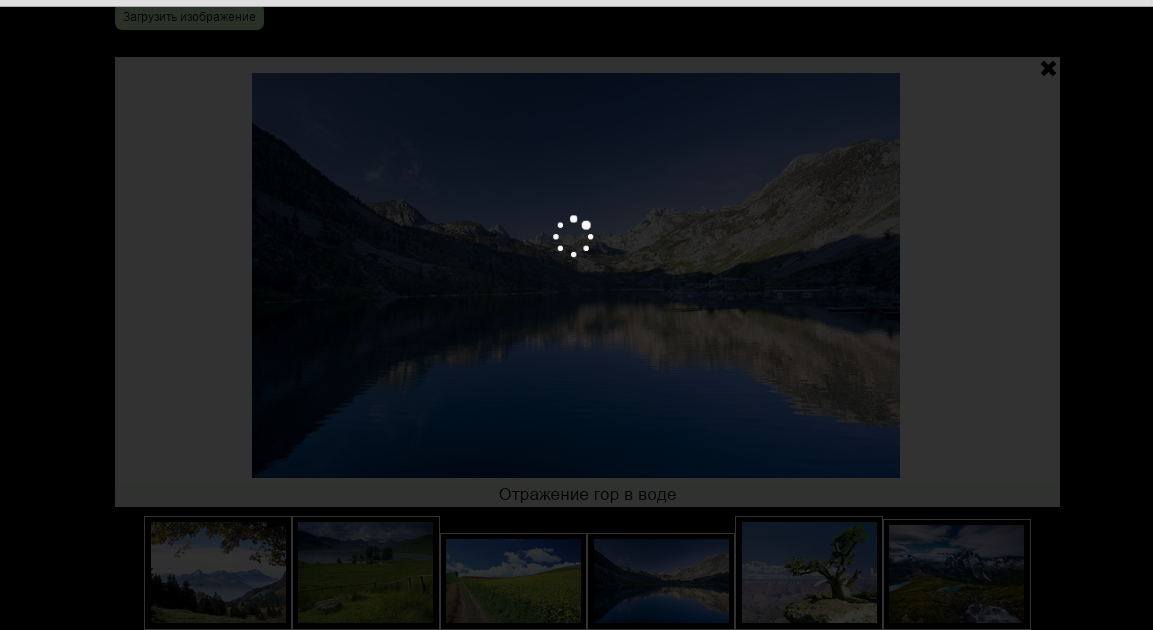


Рисунок 3 – Галерея при редактировании изображений

Приложение А. Исходные коды ИС

На листинге 1 представлен код загрузки изображений.

$UploadPath = 'Z:/home/gallery/www/images/';

$MaxSize = '20000'; # For no max size, leave blank. Max size of KBYTES.

#print "Content-type: text/html\n\n";

$File2 = param('whichfile');

$Action = param('action');

$descr=param('name');

if ($Action eq 'upload')

{

&Upload;

}

sub Upload

{#подпрограмма загрузки

#$File2 = param('whichfile');

$NewFile = "$UploadPath/$File2";

$MaxSizeBytes =1024 \* $MaxSize;

#$ok = 1;

if (!-e "$NewFile")

{#если такой файл не существует

open(OUTFILE, ">$NewFile") or &error('cant create new file of this name! $!');

binmode OUTFILE;

while ($bytesread=read($File2,$buffer,1024))

{

print OUTFILE $buffer;

}

close (OUTFILE);

$inode = stat($NewFile);

$Size = $inode->size;

if ($Size > $MaxSizeBytes)

{

# If it is too big, remove it and &error out

unlink "$NewFile" or &error("cant unlink $NewFile, too big; $!");

&error('Your file was too big!');

}

else

{

my $new = $db->prepare("INSERT INTO `photo`(`src`,`desc`) VALUES(\"/images/$File2\", \"$descr\")");

$new->execute();

}

}

}

sub error

{

my $err = shift;

print qq~

An error occured: <B>$err</B>~;

exit;

}

1;

Листинг 1 – Загрузка изображений

На листинге 2 представлен код показа галереи и миниатюр.

my $res = $db->prepare("SELECT \* FROM `photo`");

$res->execute();

my $registreted = $res->rows();

my @rows1;

if($registreted > 0)#если фото есть

{

while(my @row1 = $res->fetchrow\_array())#заполняем массив

{

push(@rows1, [@row1]);

}

}

print "<div id='one'><div id='loader'></div><img src='/images/cancel.png' id='close' alt='close' onclick='delPhoto()'/><div id='big'>";

if($registreted > 0)#если фото есть

{

if(param("src"))#если аякс передал фотографию

{

my $src=param("src");

print "<img id='image1' src=\"$src\"/>";

}

else #если никаких действий с аяксом совершено не было

{

print "<img id='image1' src='".$rows1[0][1]."'/>";

}

}

print "</div>";

if($registreted > 0)

{ if(param("desc"))#если аякс передал описание

{

my $desc=param("desc");

print "<input type='hidden' id='id' value=\"id\"><input type='text' id='desc' value=\"$desc\" onchange='editContent()' />";

}

else#если не передал(первое обращение к галерии)

{

print "<input type='hidden' id='id' value=\"$rows1[0][0]\"><input type='text' id='desc' value=\"$rows1[0][2]\" onchange='editContent()' />";

}

}

print <<main;

</div>

<div id="many">

main

my $j=0;

if($registreted > 0)

{

while($rows1[$j])

{

print "<img src='".$rows1[$j][1]."' class='photos' onclick='loadphoto(\"$rows1[$j][0]\",\"$rows1[$j][1]\",\"$rows1[$j][2]\")' />";

$j++;

}

}

print "</div>";

1;

Листинг 2 – Показ галереи изображений